# Geneva Release

## What's in/out

As of 4/15/20

### General

*To be done*

- Provide Geneva docker-compose files
- Documentation updates

*Done*

- Move to Go 1.13
- Redis as default DB (while still supporting MongoDB as alternate)
- PR Templating for improved comments on all PR submissions
- Implementation of Architectural Decision Records process and template
- Removal of export services (export client and export distro).
- Development of a common service bootstrapping module (go-mod-bootstrap) and use in all services (and SDKs).
- Removal of config-seed & implementation of service self seeding (ADR 0005)
- Deprecation of Logging service
- Deprecation of MongoDB

### Core

*To be done*

- Support for array of types
- Updates for the secure Redis deployment

*Done*

- Update current support-rules-engine (Drools) to always use local config
- Automatic Device Provisioning – backlisting support
- Addition of reading type to Event Readings (go-mod-core-contracts). Allowing for applications to have better understanding of sensor data without making a request of Metadata to get the value descriptor.
- Separation of registration and configuration modules (go-mod-registry & go-mod-configuration)
- Abstraction in go-mod-messaging to allow for 0MQ or MQTT for message communications providers (go-mod-messaging) between core data and application services. Reference implementation still uses 0MQ, but an example MQTT implementation has been documented.
- Removal of superfluous properties on the value descriptor (uomLabel, formatting)

### Device Services (and SDK)

*To be done*

- Bluetooth LE Device Service (in C). Currently under review in holding; decision on 4/20 to keep in release or have DS release after Geneva.
- ~~Visual inference device service example (in holding)~~

- Changes associated with Config Seed removal & service self-seeding (ADR 0005)
- Support for array of types
- ~~Data filter design between DS and Core Data~~

### *Done*

- Implementation of dynamic device provisioning (i.e. automatic provisioning).
- Version 2 of SDK API in C (internal API changes)
- Version 1.1.1 of SDK Go
- REST Device Service
- Camera Device Service
- Device Service BACnet in C
- Device service consistency to addres endian-ness when generating float readings in Go v. C SDKs
- Addition of type to Readings (go-mod-core-contracts). Allowing for applications to have better understanding of sensor data without making a request of Metadata to get the value descriptor. Specifically: adding "valueType" and optionally "mediaType" and "floatEncoding" to be included for floating point / binary data as appropriate
- Makefiles for C device services (making builds more consistent between Go and C device services)

## Application Services (and SDK)

### *To be done*

- JSONLogic still needs to be integrated into app service configurable.
- Batch and Send function still needs to be integrated into service configurable.
- MQTT for message communications still needs to be integrated into service configurable.
- Documentation updates (push content from local readme to edgex-docs)
- Integrate go-mod-bootstrap
- Example of JSONLogic use in app-services-examples

### *Done*

- HTTP Export use of secrets from vault
- MQTT Export use of secrets from vault
- Integration of Kuiper rules engine
- JSONLogic function (allowing for "poor man's" rules engine capability)
- Batch and Send function
- Abstraction in go-mod-messaging to allow for 0MQ or MQTT for message communications (go-mod-messaging) between core data and application services.
- Use of secret store to retrieve application service secrets.
- Store application service secrets
- Basic testing of app services
- Support for HTTPS for REST trigger
- Support MQTTS message bus trigger
- Latest core contract with type info in readings

## System Management (and CLI)

### *To be done*

- ~~CLI improvements and move out of holding~~

### *Done*

- Added blackbox tests to cover system management agent.
- Metrics and operation requests now execute concurrently (i.e. requesting metrics or start/stop/restart operations against multiple services were previously being run sequentially).

## Security

### *To be done*

- ~~Renew threat modeling~~
- ~~Ensuring the services running are those expected and authorized (w/ DevOps assistance) Has this been deferred or is it being worked?~~

### *Done*

- Improved security service bootstrapping
- Improved security testing (Blackbox tests of APIs through the API gateway)
- SIR team meeting regularly and addressing critical security issues (especially those identified by Snyk reporting)
- Services all getting secrets from Vault
- Token provider service (Create and use a per service Vault token in the security service)
- Expiration/revocation/rotation of all tokens
- Upgrade to Kong 2.0

## Dev Ops

### *To be done*

- Move Jenkins build from JJB to Jenkins Pipelines (codenamed Kraken project) – ongoing; much of this work will be done but bits remain even after the release; include in release notes

### *Done*

- Use of Advanced Synk Reporting (CommunityBridge) to detect security issues and provide vulnerability reporting; weekly reports now being generated
- Implementation of semantic versioning in build pipelines
- Provided GitHub Project Tracker, automated GitHub Issue label creation, gitcommit linter
- New life cycle policies implemented on Linux Foundation Nexus repositories
- Developer documentation now created via Pipelines
- Improved performance of builds (significantly, ARM builds now better by 15 minutes)
- Snap Global Library

## Test / QA (and documentation)

### *To be done*

- Backward compatibility testing
- Performance testing to address key performance measures **(moving to Hanoi???)**

### *Done*

- TAF tests (inclusive of setup of TAF testing architecture and framework)
- Some level of integration testing
- Basic testing of app services
- Device service blackbox testing
- Run blackbox tests in Akraino/ELIOT blueprint validated lab environment
- Documentation moved from RST to Markdown
- Documentation site has been updated and enabled with Google Analytics
- Move to Swagger documentation for all API docs (OpenAPI 3.0).  Documents all available on Swagger Hub.

## EdgeX UI

### *To be done*

### *Done*

- Combine/reduce UIs
- Inclusion of app service configurable
- Improved scheduler interface
- Removal of export services

## Certification

Deferred until LTS is available

## Vertical Solutions

### *To be done*

### *Done*

- Topcoder virtual hackathon
- China hackathon moved to 5 weeks of webinars with > 130 registrants

## Open Horizons

### *To be done*

- Development of automated build pipelines and testing

### *Done*

- Initial code contribution completed
- LF Edge as Stage 1 project